Text S1: Supporting Information

I. HOW TO USE AND RUN THE PROGRAM 2
1. Installation 2
2. Running simulation examples 2
3. Building simulations 2
3.1. Working directory 3
3.2. Using input files 4
3.3. Defining the landscape in the GUI 4
3.4. Running simulation examples 5
3.5. Defining the individuals characteristics 5
3.6. Defining the landscape changes over time 5
3.7. Defining output files 5
3.8. Running the simulation 6
II. ODD PROTOCOL 7
1. Purpose 7
2. Entities, state variables, and scales 7
2.1. Individuals 7
2.2. Grid cells 7
3. Process overview and scheduling 9
4. Design concepts 9
4.1. Basic principles 9
4.2. Emergence 9
4.3. Adaptation 9
4.4. Objectives 9
4.5. Learning 10
4.6. Prediction 10
4.7. Sensing 10
4.8. Interaction 10
4.9. Stochasticity 10
4.10. Collectives 10
4.11. Observation 10
5. Initialization 10
6. Input data 12
7. Submodels 13
7.1. Individuals related submodels 13
7.2. Initialization related submodels 15
7.3. Landscape related submodels 15
7.4. Output 15
References 16
III. EXAMPLE FILE "HABITAT_TYPE.TXT" 17

I. HOW TO USE AND RUN THE PROGRAM

1. Installation

Install NetLogo (wilensky 1999), freely available at http://ccl.northwestern.edu/netlogo/. We used NetLogo 5.0.1 to develop this simulation model, which runs under any operating systems with a Java virtual machine (Linux, MS Windows, Mac OS X).

Open NetLogo, and then open the simulation model using File / Open… / program.txt. Alternatively you can open the file "*simadapt_v20_03.txt*" using NetLogo. If the model is properly loaded, you should see "*simadapt_v20_03 – NetLogo {path to the model directory}*" at the top left of the current window.

We advise users to disable the "*view updates*" checkbox at the top of the windows which considerably slow down the simulations.

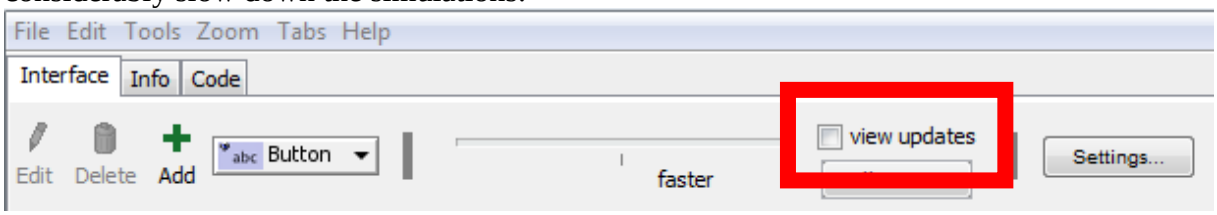
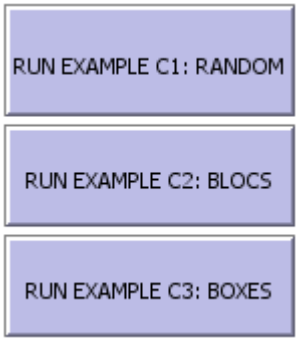


Fig. S1: Location of the "view updates" checkbox

2. Running simulation examples



The three simulation examples presented in the main text are available using the corresponding buttons "*RUN EXAMPLE C1*", "*RUN EXAMPLE C2*" and "*RUN EXAMPLE C3*". The simulations are performed once (no repetition), and take less than 40 seconds each using Windows Vista 64bits, CPU 3.06GHz, RAM 4Go. The output files are created in the directory where the program is (unless a different working directory has been specified in the input `wd`). The simulation can be repeated using the BehaviorSpace (Tools/BehaviorSpace). If your computer has multiple processor cores, then model runs could happen in parallel (number of cores defined by the user). The program can also run headless (without graphical user interface) in a distant server or a cluster of machines (see NetLogo documentation on BehaviorSpace: http://ccl.northwestern.edu/netlogo/docs/behaviorspace.html).

3. Building simulations

In this section, we described all the features of the program and parameters available in the graphical user interface (or headless using *BeahviorSpace*).

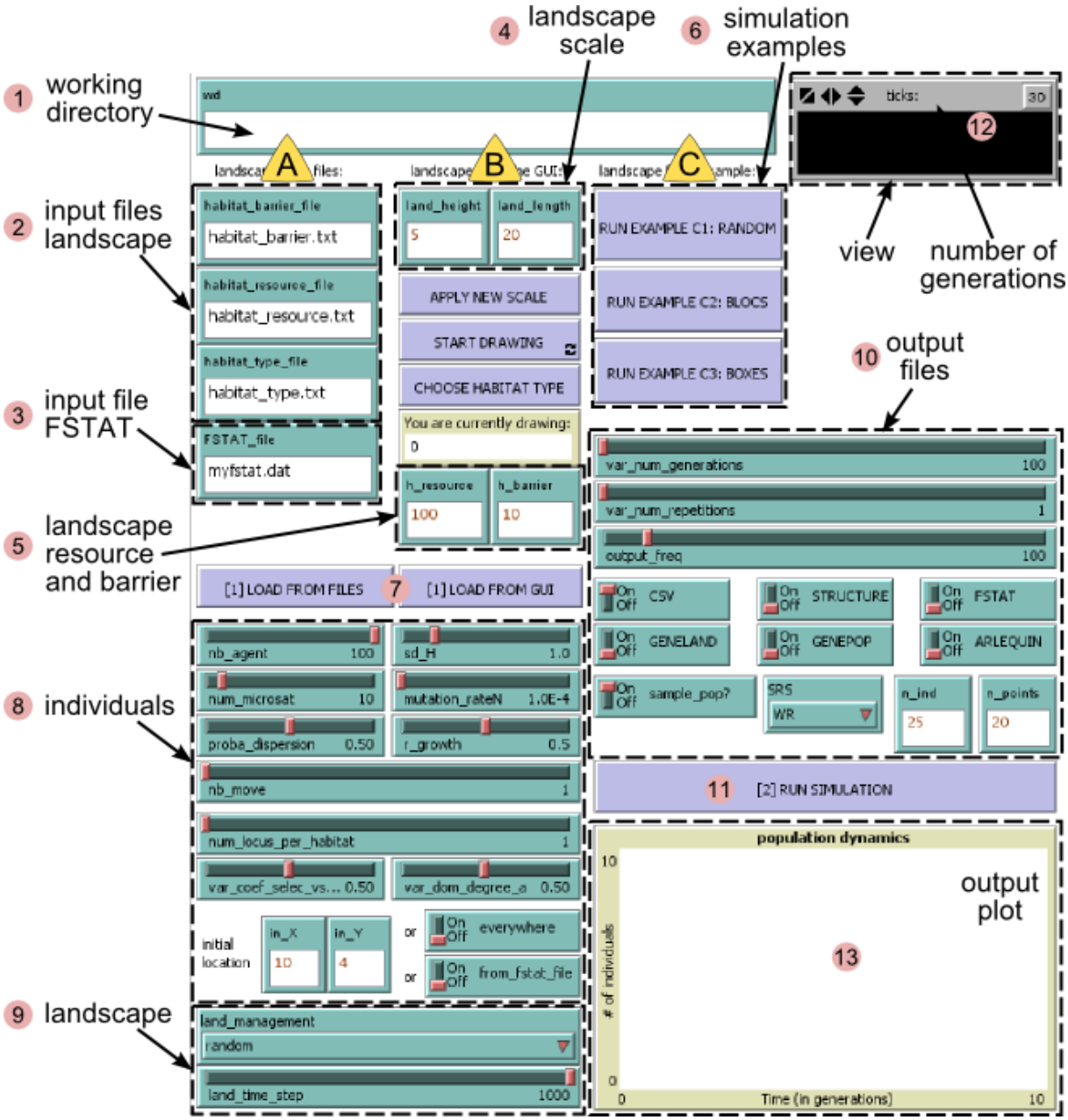


Fig. S2: Representation of the graphical user interface

3.1. Working directory

As simulations can produce a large number of files, the first step is to define the working directory `wd` (Fig. S2, box 1) where the output files will be generated. If nothing is mentioned, the current directory is used (directory of the NetLogo model file).

Using MS Windows, an example could be:
`C:\path\to\my\directory\`

Using Linux or Mac OS X:
`/home/path/to/my/directory/`

Please note that unlike the one in our example, a working directory like `C:\path\to\my\directory` will produce output files in the `C:\path\to\my\` directory with output files beginning with `directory`.

Once the working directory is defined, you can either: (Fig. S2, box A) generate a landscape from input files, (Fig. S2, box B) draw a landscape from the GUI or (Fig. S2, box C) run the simulation examples.

3.2. Using input files (Fig. S2, box A)

Landscape input files (Fig. S2, box 2) define the three layers of the landscape: the matrix of resistance for the individual dispersion (`habitat_barrier_file`), the carrying capacity for individuals (`habitat_resource_file`), and the different habitat types (`habitat_type_file`). Those files are composed of three columns: the abscissa (x from 0 to X, *i.e.* X+1 grid cells), the ordinate (y from 0 to Y, *i.e.* Y+1 grid cells) and the value of the corresponding layer. An example is provided in the third part of this supplementary material (see III. Example file *habitat_type.txt*). Files must be located in the same working directory defined earlier, and their name should include the extension *.txt*.
Once your files are created, you can load them into the model using the *"[1] LOAD FROM FILES*" button in the GUI (Fig. S2, box 7). A message will sum up the information loaded and check the correctness of the input files.
FSTAT input files (Fig. S2, box 3) allows you to load neutral markers information from an FSTAT file.

3.3. Defining the landscape in the GUI (Fig. S2, box B)

In case you want to define your own landscape files, the first thing to do is to define the scale of your landscape (Fig. S2, box 4). The scale is defined by the number of cells in abscissa (`land_lenght`), and in ordinate (`land_height`). A reasonable landscape for a simulation running on a laptop should be around 100 cells (*i.e.* 5*20 or 10*10). The "*APPLY NEW SCALE*" button updates the landscape scale (Fig. S2, box 12). At his point you should verify that the "*view updates*" checkbox is checked (see above), to visualize the landscape. You can then use the "*START DRAWING*" button.

Fig. S3: Steps during the drawing of a user-defined landscape.

Then choose (Fig. S2, box 5) the carrying capacity (`h_resource`) and resistance (`h_barrier`) associated with your first habitat type, and selects your habitat type using "*CHOOSE HABITAT TYPE*" button. Just click in the view (Fig. S2, box 12) to affect your settings to a given coordinate, and repeat the operation until all locations are set (a minimum of two habitat types is required, see Fig. 2). Finally, use the "[1] LOAD FROM GUI" button to validate your landscape. A similar text should be displayed in the Command Center:

```
Your landscape has been correctly initialized
Average landscape barrier: 10
Average landscape resource: 100
Number of habitat types: 4
```

A copy of your three input files is now located in your working directory, which can be used later as input files (*my_habitat_type.txt*, *my_habitat_barrier.txt*, *my_habitat_resource.txt*). You should now uncheck the "*view updates*" checkbox to optimize computation time.

3.4. Running simulation examples (Fig. S2, box C)

The simulation examples (Fig. S2, box 6) can be replicated using the corresponding buttons "*RUN EXAMPLE C1*", "*RUN EXAMPLE C2*" and "*RUN EXAMPLE C3*". Only one repetition is performed, which should take less than a minute in almost all computers.

3.5. Defining the individuals characteristics (Fig. S2, box 8)

`nb_agent`: number of individuals at initialization.
`in_X`: abscissa for the initial location of individuals.
`in_Y`: ordinate for the initial location of individuals.
`everywhere`: switch to ON to create `nb_agent` in every locations of the landscape.
`from_fstat_file`: switch to ON to generate individuals from an FSTAT file instead of using `nb_agent`.
`num_microsat`: number of microsatellites loci for each individual.
`sd_H`: standard deviation of the normal distribution defining the heterosygosity rate and the number of different alleles (see input section of the ODD protocol bellow).
`mutation_rateN`: probability of a stepwise mutation event.
`proba_dispersion`: probability for an individual to disperse to another location.
`nb_move`: maximum distance of dispersion for an individual (in number of cells).
`r_growth`: growth parameter of the logistic growth submodel.
`num_locus_per_habitat`: number of locus under selection for each habitat type.
`var_coef_selec_vs…`: selection coefficient against the disfavored genotype (see Hartl and jones, 1998[1]).
`var_dom_degree_a`: degree of dominance (see Hartl and jones, 1998).

3.6. Defining the landscape changes over time (Fig. S2, box 9)

`land_management`: type of land management.
`land_time_step`: frequency of landscape management (every x generations).

3.7. Defining output files (Fig. S2, box 10)

`var_num_generations`: number of generations for the simulation.
`var_num_repetitions`: number of repetitions for the simulation.
`output_freq`: frequency of output files creation (every x generations).
`CSV`: switch to create an output file of this format (never sampled).
`GENELAND`: switch to create an output file of this format.

[1] Hartl and jones, 1998. Genetics: Principles and Analysis, Jones and Bartlett Publishers

`STRUCTURE`: switch to create an output file of this format.
`GENEPOP`: switch to create an output file of this format.
`FSTAT`: switch to create an output file of this format.
`ARLEQUIN`: switch to create an output file of this format.
`sample_pop?`: switch to sampled populations (random sampling among individuals).
`SRS`: type of sampling method (WR: with replacement and WOR: without replacement)
`n_point`: number of localizations per habitat type for the sampling.
`n_ind`: number of individuals per localization for the sampling.

3.8. Running the simulation

Once the landscape and the individuals are set, the simulation can run using the *"[2] RUN SIMULATION*" button (Fig. S2, box 11). The population dynamics is shown in plot (Fig. S2, box 13).

II. ODD PROTOCOL

The model description follows the ODD (Overview, Design concepts, Details) protocol for describing individual- and agent-based models (Grimm *et al.* 2006, 2010). It is destined to readers looking for a full description and verification of a particular process, but most of all for users willing to modify or extend the code to their own study cases.

1. Purpose

Worldwide, populations evolve on various habitats, in which the level of modification or alteration by human activities is heterogeneous. The aim of this model is to study the impact of human activities, through land use, on the structure of populations in order to establish scenarios including changes likely to append in land use or at a broader context, global changes.

2. Entities, state variables, and scales

2.1. Individuals

Each individual in the model represents an entity, characterized by state variables updated at each time step (see Table S1) stored in an object `turtle` located in a given grid cell (landscape). Each time step in the model corresponds to one individual generation (non-overlapping) and simulations could run for any number of time steps.

Table S1. State variables for individuals.

State variable names in the code	*Definition*	*Type*
`self`	Number to identify each individual	String
`cap_move`	Default value for individuals dispersion capabilities	Integer
`markerN`	List of neutral markers (microsatellites)	List (Integer)
`markerS`	List of alleles for each locus under selection	List (Integer)
`generation`	Variable to separate generations (“old” or “new”)	String
`wi`	Fitness	Float
`mymum`	Genealogy	String
`mydad`	Genealogy	String
`sample`	Variable used in sampling (with or without replacement)	Boolean

2.2. Grid cells

The grid represents the landscape in which the individuals evolve. Each cell is characterized by its state, descriptor of environmental conditions which drive the behavior and dynamics of individuals, composed of a barrier variable representing the cost to move to a given cell (`habitat_barrier`), a resource representing the suitability of a given patch (`habitat_resource`), i.e. the carrying capacity and a habitat type representing the characteristics regarding adaptation (`habitat_type`). The cells changes over time according to management rules. The landscape can be defined in the Graphical User Interface (GUI), see Fig. S4.

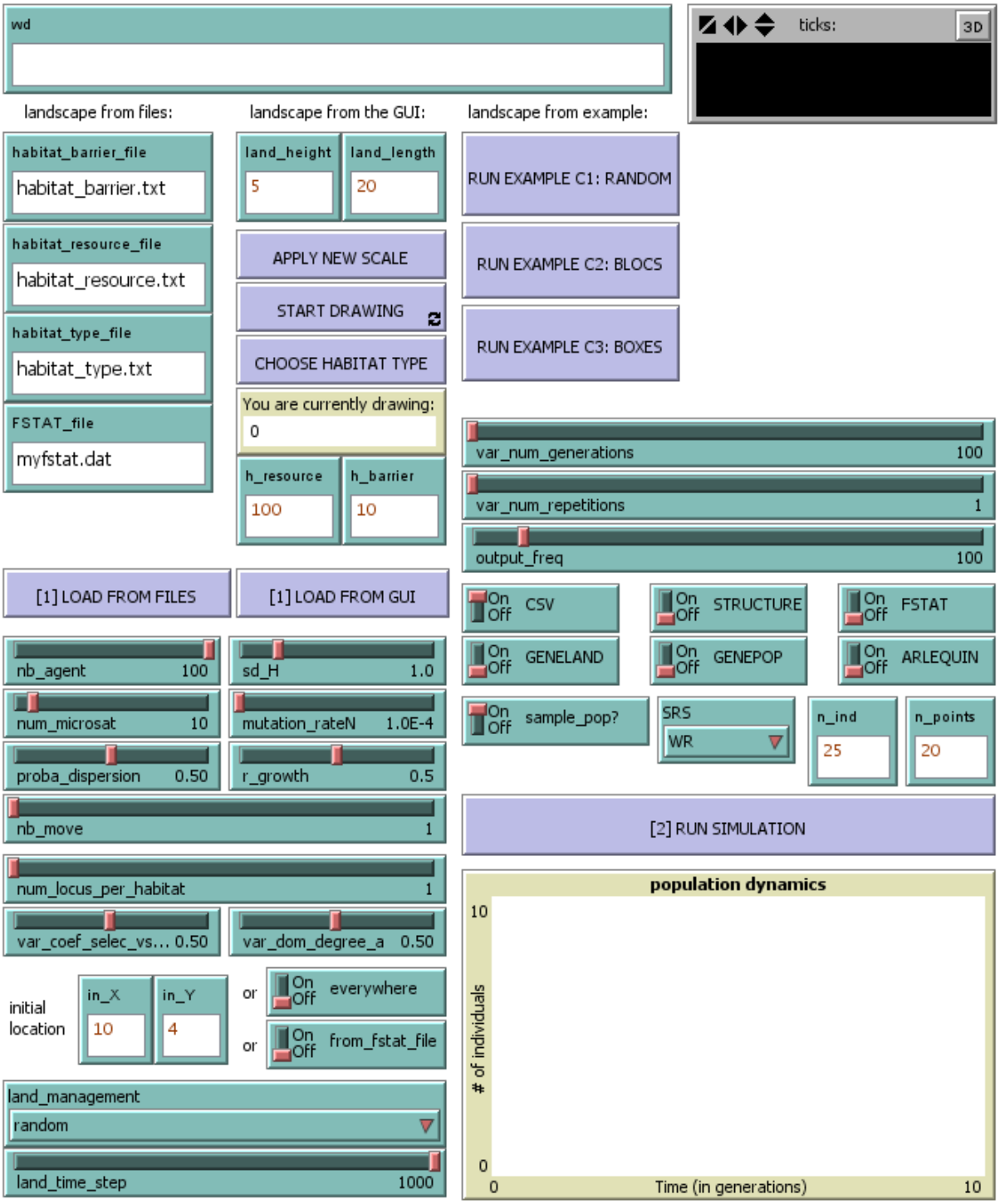


Fig. S4: Graphical User Interface of the simulation model using NetLogo 5.0.1 (wilensky 1999)

2.3. Collectives

Group of individuals located in the same habitat type (or in the same location), are considered as populations for further analysis.

3. Process overview and scheduling

The model processes include the dispersion and reproduction of individuals in this given order for each given individual taken in a random order. Time is modeled as discrete steps. At the end of one time step, all individuals die, and the next time step begins with their offspring (*i.e.* non-overlapping generations). The information regarding the previous generation after reproduction is stored in an output file only if requested. All individuals' state variables are updated asynchronously as they are taken in a random order.

4. Design concepts

4.1. Basic principles

The model design is based on previous work on the fields of landscape genetics where significant advances have been made on the last decade (Manel *et al.* 2003 ; Manel and Segelbacher 2009). This simulation model is generic enough to be adapted for all living forms and is based on previous works by Gavrilets *et al.* 2007 and Gravilets and Vose, 2005 for their study on *Cichlidae* ; Duenez-Guzman *et al.* 2009 for their work on *Heliconius* ; Bruggeman *et al.* 2010 and Bruggeman *et al.* 2009 for their work on *Picoides borealis* ; Philips *et al.* 2004 for their work on trees ; Lawson et Jensen, 2005 ; Saledin et Littlejohn, 2003 ; Landguth *et al.* 2010a ; Landguth *et al.* 2010b ; Landguth *et al.* 2010c and Jaquiéry *et al.* 2011 for their generic studies. Our work differ from those for its generic approach with genetic, ecological and landscape submodels with a high level of abstraction allowing to attempt the construction of scenarios rather than explaining a given situation (from real world to theoretical situations). The landscape submodel can run asynchronously with others submodels to reproduce time lags or independent landscape management. Our work differs from other forward-time simulation program such as simuPop (Peng and Kimmel 2005), Nemo (Guillaume and Rougemont 2006) or quantiNEMO (Neuenschwander *et al.* 2006) for its focus on landscape genetics with a highly flexible and extendable landscape submodel linked to the population genetics approach.

4.2. Emergence

Almost all results of the model emerged from the behavior of the individuals depending on their adaptive traits in the given changing landscape and were expected to vary in complex ways when particular characteristics of individuals or their environment changed. However initial characteristics of individuals are imposed and hence dependent on what type of individual is simulated, and hence 'built in' rather than emergent results.

4.3. Adaptation

Adaptive traits were considered for all individuals through reproduction. Individuals with a genetic background in favor of a specific habitat are expressing a better fitness than other individuals. Applying Mendel inheritance laws and ecology concept of carrying capacity, the next generation have a higher proportion of individuals with high fitness.

4.4. Objectives

Individuals do not make decisions by ranking alternatives and fitness is a consequence of random move and genetic background. Individuals' objective is to reproduce, which results in the colonization of the landscape.

4.5. Learning

Individuals change their adaptive traits over time as they adapt to a given habitat. These changes occur between generations and not during a time step (*i.e.* no learning). The adaptation is heritable and subject to mutations.

4.6. Prediction

The dispersion of individuals can be considered as a tacit prediction that dispersion to another cell will give to the individuals a better probability of survival.

4.7. Sensing

Individuals perceive their peers in a given cell and the location of potential destinations in their neighborhood (used for dispersion). The mechanisms by which individuals obtain information are modeled explicitly.

4.8. Interaction

Interactions between individuals are direct for reproduction and indirect for mediating resources (carrying capacity). They are no communication between individuals.

4.9. Stochasticity

The mating of individuals in a given cell are modeled randomly to reproduce variability and because we assumed that actual causes of the variability were unimportant given the model objectives. The dispersion of individuals is random among possible destinations because we assumed that individuals do not have a perception of habitats located in other cells.

4.10. Collectives

Individuals in a given habitat (or a given location) belong to an aggregate named population. Population is the result of adaptation and dispersion. Population genetics analyses are made at these defined levels (habitat type or location).

4.11. Observation

All information regarding individuals can be collected for analyzing the model after each reproduction process (*i.e.* at each time step). This information can also be sampled and used to imitate what can be observed in an empirical study (see the "Virtual Ecologist" approach by Zurell *et al.* 2010, as mentioned by Grimm *et al.* 2010).

5. Initialization

The initial state of the model (*i.e.* at time $t = 0$) is composed of a user-defined number of individuals located in the landscape. Alleles at both the neutral loci and loci under selection are initialized randomly according to the setting and consequently vary among simulations. In this simulation model, consequences of initial state are studied so that user-defined initialization is of importance in order the results to be accurately replicated. Default values for the simulation examples are provided in Table S2

Table S2. Default values at initialization.

Variable names	*Definition*	*Default value*
INIT.		
wd	Working directory	
nb_agent	Number of individuals per cell	100
in_X	X Coordinate of individuals	10
in_Y	Y Coordinate of individuals	4
everywhere	To specify that individuals are located in every cells	False
from_fstat_file	To specify if individuals are created according to an input file	False
Sd_H	Standard deviation of the normal distribution defining the number of alleles and heterozygosity among the population	1
LANDSCAPE		
land_management	Type of land management	None
land_time_step	Time steps for land management	1000
habitat_type_file	Name of the file containing habitat types	"*habitat_type.txt*"
habitat_barrier_file	Name of the file containing habitat barriers	"*habitat_barrier.txt*"
habitat_resource_file	Name of the file containing habitat resources	"*habitat_resource.txt*"
INDIVIDUALS		
num_microsat	Number of microsatellites loci	10
num_locus_per_habitat	Number of loci under selection per habitat type	1
proba_dispersion	Dispersion rate	0.50
nb_move	Number of movements	1
r_growth	Growth parameter of the logistic growth submodel	0.5
var_coef_selec_vs_xx	Selection coefficient against the deleterious genotype	0.5
var_dom_degree_a	Degree of dominance of the deleterious allele	0.5
mutation_rateN	Mutation rate	10^{-4}
OUTPUTS		
output_freq	Outputs frequency (every *x* time steps)	100
var_num_generations	Number of generations	100
var_num_repetitions	Number of repetitions	1
CSV	Comma-separated values output file	True
GENEPOP	Genepop output file	False
FSTAT	Fstat output file	False
ARLEQUIN	Arlequin output file	False
STRUCTURE	Structure output file	False
GENELAND	Geneland output file	False
sample_pop?	Sampling of individuals	True
n_points	Number of points per habitat type	20
n_ind	Number of individuals per point sampled	25
SRS	To specify if the sampling is made with (WR) or without (WOR) replacement	WR

6. Input data

The simulation model can use external files for the integration of the landscape layers (GIS). These files concern the landscape habitat ("*habitat_type.txt*"), the landscape resource ("*habitat_resource.txt*") and the landscape barrier ("*habitat_barrier.txt*"). Fig. S5 represents an example of input file at initialization for habitat types. If no files are provided, the model will run with default values of 1, 100 and 10, respectively. Each file is organized with coordinates in the first two columns (abscissa, ordinate) and landscape characteristic in the third as following (see also section III):

0	0	1
1	0	4
2	0	1
…		
10	10	3

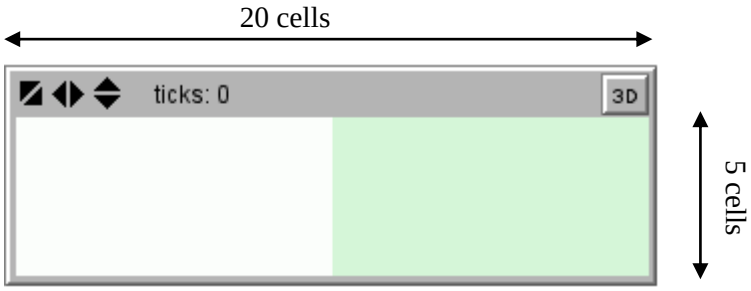


Fig. S5: Graphical representation of an example of habitat types at initialization

Alleles are chosen in a normal distribution with a user-defined standard deviation `sd_H` corresponding to an expected number of alleles and rate of heterozygosity (See Fig. S6). For example, when `sd_H` = 1, the expected heterozygosity is around 70% and the number of possible alleles is around 10.

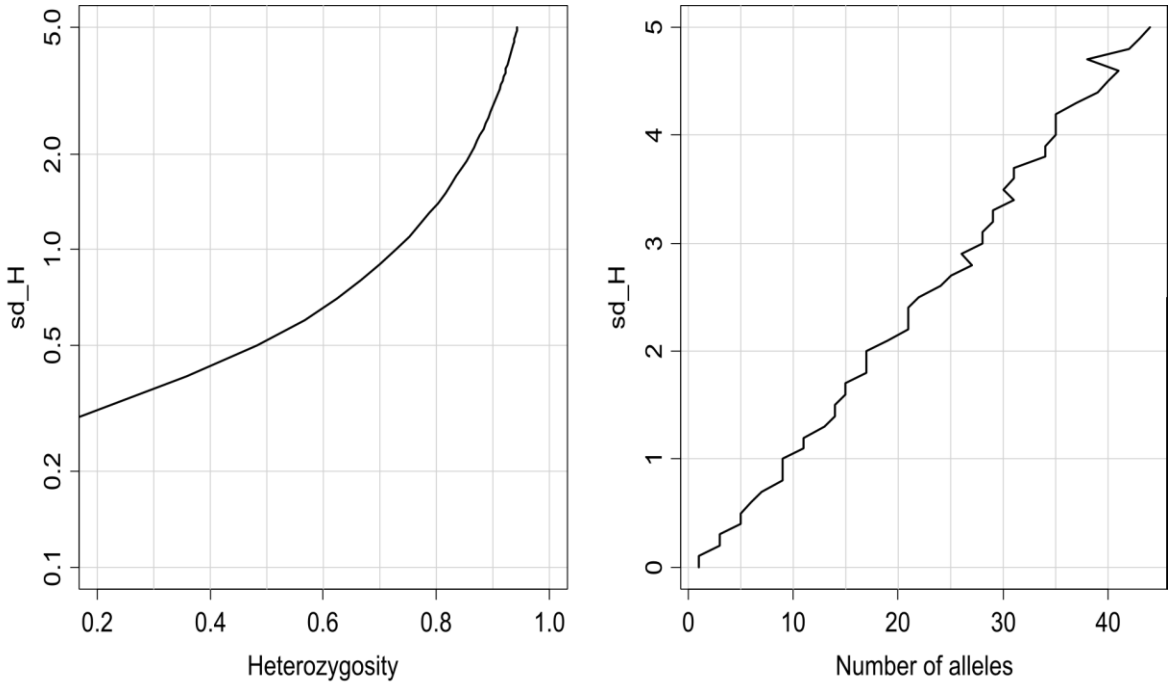


Fig.S6: Heterozygosity and number of alleles corresponding to the parameterization of `sd_H` in the GUI, using 10000 random values in a normal distribution

7. Submodels

7.1. Individuals related submodels

7.1.1. Reproduction (`main_reproduction`)

Description

The reproduction submodel is based on the assumption that all individuals located in a given cell have a probability of mating, and that mating occurs as if all gametes were chosen in the population (*i.e.* gamete pool) with a probability defined by individuals' selective values. Then transmission of genetic characteristics follows Mendel inheritance laws. The number of descendants is defined by the logistic growth submodel according to fitness traits depending on the habitat type (`habitat_type`). The population genetics submodel is detailed in the main text and not copied in this supplementary material. Mutations occur only in microsatellites loci with a probability of 10^{-4} which can be modified through the GUI.

Validation

The population genetics submodel is based on Hartl (2005), Wade *et al.* (2001) and Trajstman (1973). We assumed a stepwise mutation model as defined by Hamilton (2009, p 169).

Verification

In the code the selective values w_i of each individual is stored in a list (`allwi` in procedure `main_reproduction`). Then two gametes are chosen in the gamete pool with a probability based on w_i so that selection is conserved. The selection of two gametes is reproduced *f* times for the creation of the next generation. This submodel has been checked by verifying that selection was operational over a large number of individuals with a broad range of selection coefficients. To verify that Mendel inheritance laws were adequately reproduced by our model, we simulated more than 1000 first generations with a marked gamete and checked the allele frequencies of 10 loci (allele frequency of 0.49995, CI95% = [0.4952743 ; 0.5046257]).

```
Pseudo-algorithm
FOR all cells {
     IF at least 2 individuals {
          define number of offspring Nt1 from Nt0
          REPEAT Nt1 times {
               choice into gamete pool (probability wi)
               create descendant with Mendel inheritance laws
          }
     }
     remove old generation of individuals
}
```

7.1.2. Survival (`main_reproduction`)

Description

Each cell in the grid is characterized by a carrying capacity defining the maximum number of individuals that a cell can contain (variable `habitat_resource`). The number of individuals is controlled by a logistic growth submodel.

Verification
This submodel can be checked at any moment through the GUI using the population dynamics plot which monitors the number of individuals as a function of time.

7.1.3. Dispersion (`main_dispersion`)

Description
Each individual can move from one cell to another located in its Moore neighborhood. Each cell is characterized by its resistance (patch variable `habitat_barrier`). Lowest values of resistance represent easy to cross cells and highest values impermeable cells. On the other side, each individual is characterized by its dispersion capabilities (individual variable `cap_move`). Consequently, an individual with higher dispersion capabilities could move to more cells within its Moore neighborhood (see Fig. S7). The decision whether to move to another cell is based on a fixed probability (default probability of 0.5). The destination cell is chosen randomly among potential destinations.

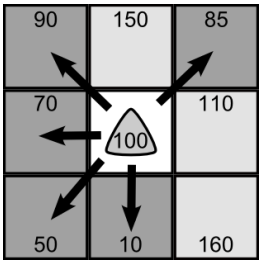


Fig. S7. An individual is located in the central cell with a dispersion capability of 100. Potential destinations are represented in dark grey.

Validation
We assumed that individuals do not have a perception of the suitability of neighboring cells and disperse randomly into the landscape.

Verification
This submodel has been checked by verifying that the number of individuals selected for dispersal corresponded to the rate of dispersion, by verifying that the potential destination cells were adequate to dispersal capabilities, that the individuals moving were properly removed from the buffer list, and finally that individuals were properly relocated to chosen destinations.

Pseudo-algorithm
```
FOR all individuals {
   WHILE individual can move {
      list of potential destinations
      move to a random cell among potential destinations
      update of dispersal capacity according to movement
   }
}
```

7.2. Initialization related submodels

7.2.1. Setting up the list of alleles for neutral and under selection loci

Each allele for microsatellites loci is chosen in a normal distribution of standard deviation `sd_H` to obtain a given heterozygosity and number of possible alleles. Alleles at loci under selection are chosen randomly at initialization.

7.2.3. Setting up the individuals

Identification numbers are incremented during the reproduction process and for the simulations up to 100 generations, and then reinitiated at each generaion. Number of alleles and heterozygosity are presented in “input” section.

7.3. Landscape related submodels

7.3.1. Landscape generation

External files:

- “*habitat_type.txt*” for the landscape habitat type,
- “*habitat_resource.txt*” for the landscape carrying capacity and
- “*habitat_barrier.txt*” for the landscape resistance.

In case files contain errors, a default value is attributed to each layer.

7.3.2. Landscape management

Landscape can change over time due to human actions. The simulation model includes a landscape management submodel that reproduces various scenarios.
Alternatives are:

- "none"	no landscape management
- "random"	random changes over time
- "neighbor4"	changes to one of the 4 neighbors
- "neighbor8"	changes to one of the 8 neighbors
- "transition"	changes to one of the existing habitat type
- "user-defined"	to let the user program his own submodel

7.4. Output

Description
In order to perform population genetics analysis, the simulation model produces output files for the most-used population genetics software (see main text). The sampling method is based on random location and random individuals.

Validation
The choice of software was realized following Excoffier and Heckel 2006.

Verification
Each output has been checked using the corresponding software.

References

Bruggeman D.J., Jones M. L., Scribner K.T., Lupi, F., 2009. Relating Tradable Credits for Biodiversity to Sustainability Criteria at a Landscape-Scale. Landscape Ecology 24, 775-790.

Bruggeman D.J., Wiegand T., Fernández N., 2010. The relative effects of habitat loss and fragmentation on population genetic variation in the red-cockaded woodpecker (Picoides borealis). Molecular Ecology 19, 3679-3691.

Duenez-Guzman E.A., Mavarez J., Vose DM.D., Gavrilets S., 2009. Case studies and mathematical models of ecological speciation. 4. Hybrid speciation in butterflies in a jungle. Evolution 63, 2611-2626.

Excoffier L and Heckel G (2006) Computer programs for population genetics data analysis: a survival guide. Nature Reviews Genetics 7, 745-758.

Gavrilets S., Vose A., 2005. Dynamic patterns of adaptive radiation. PNAS 102, 18040-18045.

Gavrilets S., Vose A., Barluenga M., Salzburger W., Meyer A., 2007. Case studies and mathematical models of ecological speciation. 1. Cichlids in a crater lake. Molecular Ecology 16, 2893-2909.

Grimm V., Berger U., Bastiansen F., Eliassen S., Ginot V., Giske J., Goss-Custard J., Grand T., Heinz S., Huse G., Huth A., Jepsen JU., Jørgensen C., Mooij WM., Müller B., Pe'er G., Piou C., Railsback SF., Robbins AM., Robbins MM., Rossmanith E., Rüger N., Strand E., Souissi S., Stillman RA., Vabø R., Visser U., DeAngelis DL., 2006. A standard protocol for describing individual-based and agent-based models. Ecological Modelling 198, 115-126.

Grimm V., Berger U., DeAngelis DL., Polhill G., Giske J., Railsback SF., 2010. The ODD protocol: a review and first update. Ecological Modelling 221, 2760-2768

Guillaume F, Rougemont J (2006) Nemo: an evolutionary and population genetics programming framework. Bioinformatics, 22, 2556-2557.

Hamilton M.B., 2009. Population genetics. Eds John Wiley & Sons, 407pp.

Hartl DL (2005) A primer of population genetics, eds. Sinauer Associates.

Holderegger R., Wagner H.H., 2006. A brief guide to landscape genetics. Landscape Ecology 21, 793-796.

Jaquiéry J., Broquet T., Hirzel A.H., Yearsley J., Perrin N., 2011. Inferring landscape effects on dispersal from genetic distances : how far can we go ? Molecular Ecology 20, 692-705.

Landguth E.L., Cushman S.A., 2010b. CDPOP : A spatially explicit cost distance population genetics program. Molecular Ecology Resources 10, 156,161.

Landguth E.L., Cushman S.A., Murphy M.A., Luikarts G., 2010a. Relationships between migration rates and landscape resistance assessed using individual-based simulations. Molecular Ecology Resources 10, 854,862.

Landguth E.L., Cushman S.A., Schwartz M.K., McKelvey K.S., Murphy M., Luikart G., 2010c. Quantifing the lag time to detect barriers in landscape genetics. Molecular Ecology 19, 4179-4191.

Lawson D., Jensen H.J., 2006. The species-area relationship and evolution. Journal of Theoretical Biology 241, 590-600.

Manel S., Schwartz M.K., Luikart G., Taberlet P., 2003. Landscape genetics : combining landscape ecology and population genetics. Trends in Ecology and Evolution 18, 189-197.

Manel S., Segelbacher G., 2009. Perspectives and challenges in landscape genetics. Molecular Ecology 18, 1821,1822.

Neuenschwander S, Hospital F, Guillaume F, Goudet J (2008) quantiNEMO: an individual-based program to simulate quantitative traits with explicit genetic architecture in a dynamic metapopulation. Bioinformatics, 24, 1552-1553.

Peng B, Kimmel M (2005) simuPOP: a forward-time population genetics simulation environment. Bioinformatics, 21, 3686-3687.

Phillips P.D., Thompson I.S., Silva J.N.M., van Gardingen P.R., Degen B., 2004. Scaling up models of tree competition for tropical forest population genetics simulation. Ecological Modelling 180, 419-434.

Saladin S., Littlejohn M., 2003. A spatially explicit individual-based model of reinforcement in hybrid zones. Evolution 57, 962-970.

Trajstman A.C., 1973 The necessity of the poisson distribution for the equivalence of some random mating models. Mathematical Biosciences 17, 1-10.

Wade MJ, Winther RG, Agrawal AF, Goodnight CJ (2001) Alternative definitions of epistasis : dependence and interaction. TRENDS in Ecology and Evolution 9, 498-504.

Wilensky U (1999) NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL.

Zurell D., Berger U., Cabral J. S., Jeltsch F., Meynard C. N., Münkemüller T., Nehrbass N., Pagel J., Reineking B., Schröder B., Grimm V., 2010. The virtual ecologist approach: simulating data and observers. Oikos, 119: 622–635. doi: 10.1111/j.1600-0706.2009.18284.x

III. EXAMPLE FILE "HABITAT_TYPE.TXT"

This file is made of three columns: the abscissa (x from 0 to 19, *i.e.* 20 grid cells), the ordinate (y from 0 to 4, *i.e.* 5 grid cells) and the habitat type (from 1 to 2, *i.e.* 2 habitat types).

```
0	0	1
1	0	1
2	0	1
3	0	1
4	0	1
5	0	1
6	0	1
7	0	1
8	0	1
9	0	1
10	0	2
11	0	2
12	0	2
13	0	2
14	0	2
15	0	2
16	0	2
17	0	2
18	0	2
19	0	2
0	1	1
1	1	1
2	1	1
3	1	1
4	1	1
5	1	1
6	1	1
7	1	1
8	1	1
9	1	1
10	1	2
11	1	2
12	1	2
13	1	2
14	1	2
15	1	2
16	1	2
17	1	2
18	1	2
19	1	2
0	2	1
1	2	1
2	2	1
3	2	1
4	2	1
5	2	1
6	2	1
7	2	1
8	2	1
9	2	1
10	2	2
11	2	2
12	2	2
```

13	2	2
14	2	2
15	2	2
16	2	2
17	2	2
18	2	2
19	2	2
0	3	1
1	3	1
2	3	1
3	3	1
4	3	1
5	3	1
6	3	1
7	3	1
8	3	1
9	3	1
10	3	2
11	3	2
12	3	2
13	3	2
14	3	2
15	3	2
16	3	2
17	3	2
18	3	2
19	3	2
0	4	1
1	4	1
2	4	1
3	4	1
4	4	1
5	4	1
6	4	1
7	4	1
8	4	1
9	4	1
10	4	2
11	4	2
12	4	2
13	4	2
14	4	2
15	4	2
16	4	2
17	4	2
18	4	2
19	4	2